09 Verification of Valid ISBNs

Most books now published are assigned a code which uniquely identifies the book. The International Standard Book Number , or ISBN, is normally a sequence of 10 decimal digits, but in some cases, the capital letter X may also appear as the tenth digit. Hyphens are included at various places in the ISBN to make them easier to read, but have no other significance. The sample input and expected output shown below illustrate many valid, and a few invalid, forms for ISBNs.

Actually, only the first nine digits in an ISBN are used to identify a book. The tenth character serves as a check digit to verify that the preceding 9 digits are correctly formed. This check digit is selected so that the value computed as shown in the following algorithm is evenly divisible by 11. Since the check digit may sometimes need to be as large as 10 to guarantee divisibility by 11, a special symbol was selected by the ISBN designers to represent 10, and that is the role played by X.

The algorithm used to check an ISBN is relatively simple. Two sums, *s1* and *s2*, are computed over the digits of the ISBN. *s1* is the partial sum of the digits of ISBN and *s2* the partial sum of *s1*. The ISBN is correct if the final value of *s2* is evenly divisible by 11. Some writers use the term *cummulative sum* instead of *partial sum*, but they both mean the same thing.

An example will clarify the procedure. Consider the (correct) ISBN 0-13-162959-X. First look at the calculation of *s1*:

digits in the ISBN	0	1	3	1	6	2	9	5	9	10 (X)
s1 (partial sums)	0	1	4	5	11	13	22	27	36	46

The calculation of *s2* is done by computing the total of the partial sums in the calculation of *s1*:

s2 (partial sums of s1)	0	1	5	10	21	34	56	83	119	165

We now verify the correctness of the ISBN by noting that 165 is, indeed, evenly divisible by 11.

The input ISBN will be in a file called isbn.txt. In that file there will be one ISBN per line. The ISBNs may be valid or invalid. The ISBNs may have zero or more hyphens ("-"). The invalid ISBNs may have extraneous characters in them or may have insufficient number of digits. Your program should be general enough to process any file in the format that we just discussed and not just on the sample file that we have linked to. The input file that we will be testing your program on will be different from the one provided.

Your program will open the file *isbn.txt* for reading. You will read one line at a time as a string, then parse the string character by character and store the digits and the character *X* into a list. Remember the hypen ('-') character can occur anywhere in the string. There are several tests that you will have to perform to insure that you have a valid ISBN.

- Only the characters '0' through '9', 'X' or 'x', and '-' is present in the input string.
- There are exactly nine digits from '0' through '9' and the last character is either a digit or 'X' or 'x'.

You will design your program having several functions. You will also use some of the functions in the *String* module. You will create two lists *s1* and *s2*that will hold the partial sums as outlined above. If the last element in the array *s2* is divisible by 11 then you have a valid ISBN.

You will open another file *isbnOut.txt* for writing. After you have read a line from the input file and determined whether it is a valid ISBN or not you will write out the result in the output file. The format will be as follows.

```
0-1315-2447-X  valid
0-89237-010-9  invalid
```

Close the files *isbn.txt* and *isbnOut.txt* after you have finished processing.